Data Management and Business Intelligence - Assignment 1

Academic Year: 2018-2019 (Full-Time)
Assignment Partners: Baratsas Sotiris - f2821803 | Spanos Nikos - f2821826

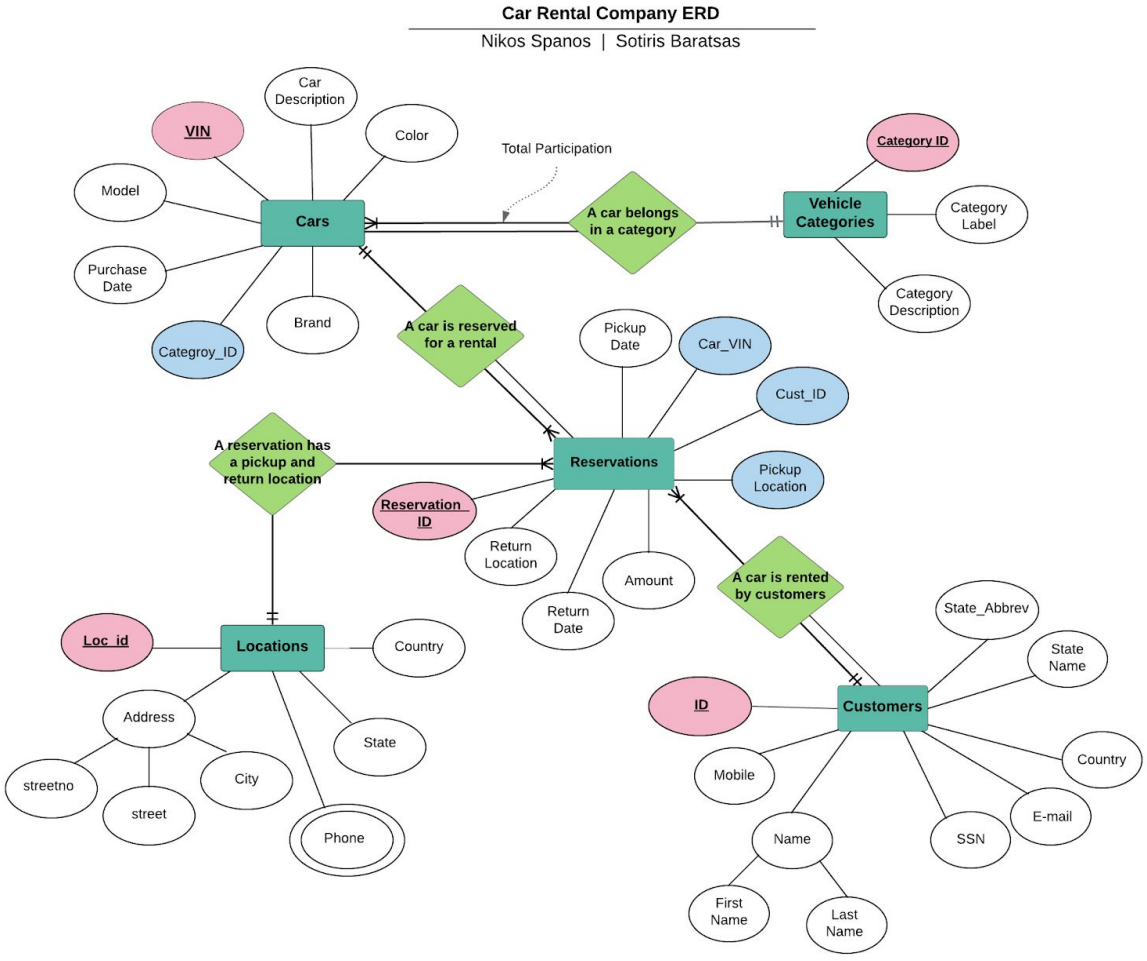
Including this report, the deliverables of this assignment are:

1) Report.pdf - A concise report including assumptions made and actions taken.
2) erd.png - Entity-Relationship-Diagram (ERD)
3) dbmodel.png - Relational Database Schema
4) create.sql - Create Statements for our schema
5) insert.sql - Insert Statements for our data
6) queries.sql - Answers to the assignment questions in the form of MySQL queries
7) /task4 - Includes an **.Rd** file for questions 4 and a copy of the .csv file
8) /task5 - Includes an **.Rd** file for question 5

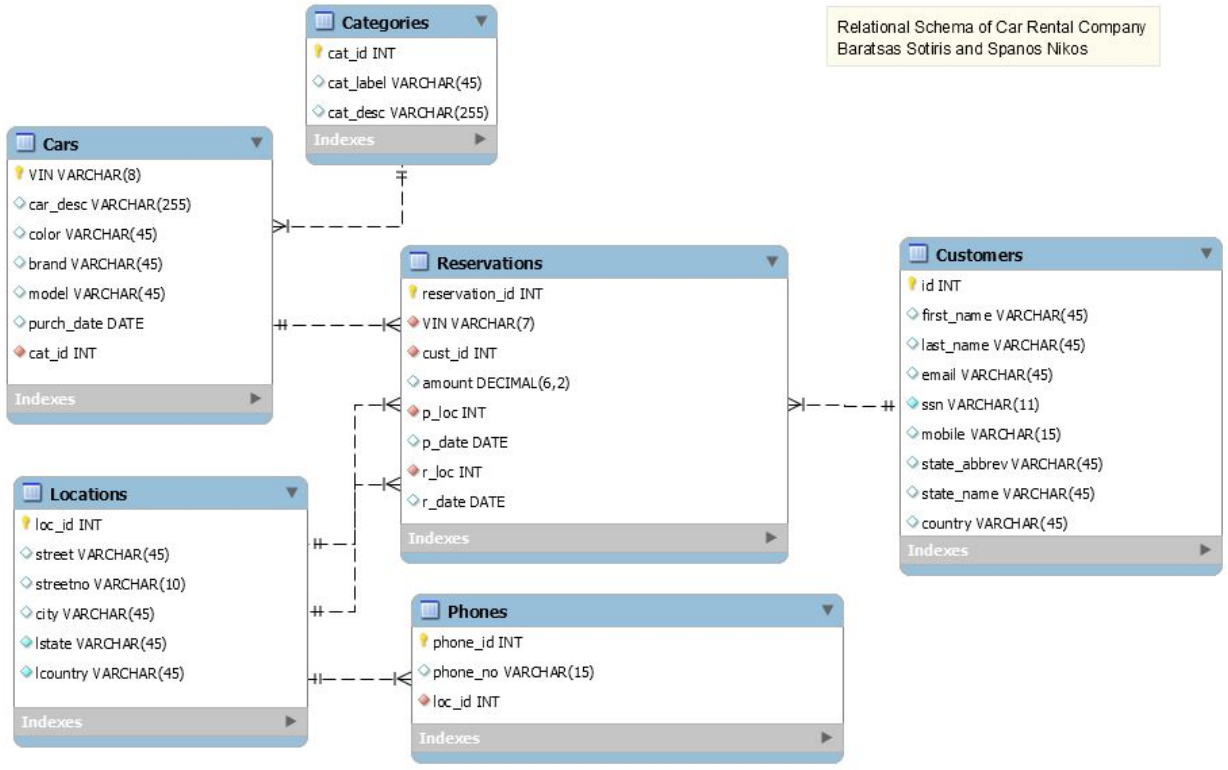
Section 1: Entity Relationship Diagram - Assumptions

Our ERD model contains **five** distinct Entities, **four** Relationships and the respective Attributes. Specifically:

Entities: Cars, Vehicle Categories, Customers, Locations and Reservations.
Entities are indicated with blue color.
Relationships are indicated with green color.
Primary Keys are indicated with pink color.
Foreign Keys are indicated with light blue color.



Section 2: Relationship Schema Model - Assumptions



Cardinalities:

• Cars – Vehicle Categories: many to one (∞ → 1).

• Cars - Reservations: one to many (1 → ∞).

Assumption 1 :A car can be rented many times or never be rented at all, **but** a reservation number includes only one car at a specific date range {pickup date, return date}. Total Participation of Cars to rentals.

• Reservations - Customer: many to one (∞ → 1).

Assumption 2: The same customer can have more than one reservations on different dates {pickup date, return date} with different cars.

• Cars - Reservations - Customers: one to one (1 → 1).

Assumption 3: A car is rented by only one customer. If the same customer wants to rent a second rental will need to issue a new reservation number. Total Participation of Customers.

• Locations - Reservations: one to many (1 → ∞).

Assumption 4: A rental belongs to no more that one locations. Rental's location is identified by his pickup location when the reservation number was issued.

----------END OF SECTION 2----------

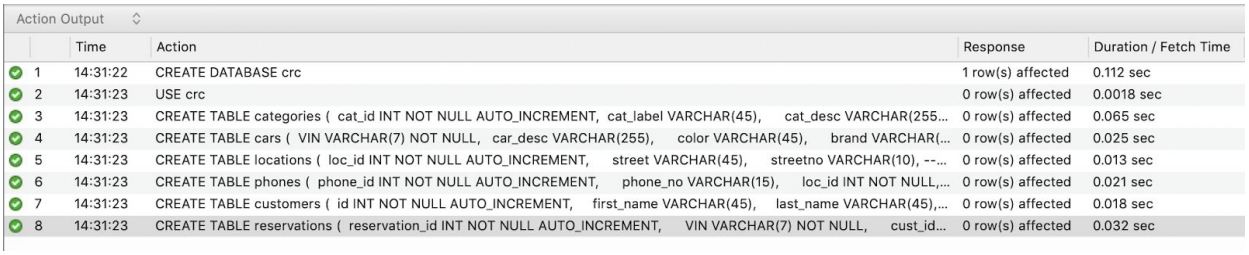
Section 3: Create Statements

CODE

```
CREATE DATABASE crc;
USE crc;

CREATE TABLE categories (
	cat_id INT NOT NULL AUTO_INCREMENT,
	cat_label VARCHAR(45),
	cat_desc VARCHAR(255),
	PRIMARY KEY (cat_id)
);

CREATE TABLE cars (
	VIN VARCHAR(7) NOT NULL,
	car_desc VARCHAR(255),
	color VARCHAR(45),
	brand VARCHAR(45),
	model VARCHAR(45),
	cat_id INT,
	purch_date DATE,
	PRIMARY KEY (VIN),
	FOREIGN KEY (cat_id) REFERENCES categories(cat_id)
);

CREATE TABLE locations (
	loc_id INT NOT NULL AUTO_INCREMENT,
	street VARCHAR(45),
	streetno VARCHAR(10), -- We put it Varchar to account for cases like "34A
Houston street"
	city VARCHAR(45),
	lstate VARCHAR(45),
	lcountry VARCHAR(45),
	PRIMARY KEY (loc_id)
);

CREATE TABLE phones (
	phone_id INT NOT NULL AUTO_INCREMENT,
	phone_no VARCHAR(15),
	loc_id INT NOT NULL,
	PRIMARY KEY (phone_id),
	FOREIGN KEY (loc_id) REFERENCES locations(loc_id)
);

CREATE TABLE customers (
	id INT NOT NULL AUTO_INCREMENT,
	first_name VARCHAR(45),
	last_name VARCHAR(45),
	email VARCHAR(45),
	ssn VARCHAR(11) NOT NULL, -- We put it as Varchar(11) to accommodate a 9
digit number with 2 dashes.
	mobile VARCHAR(15),
	state_abbrev VARCHAR(45),
	state_name VARCHAR(45),
	country VARCHAR(45),
	PRIMARY KEY (id)
);

CREATE TABLE reservations (
	reservation_id INT NOT NULL AUTO_INCREMENT,
	VIN VARCHAR(7) NOT NULL,
	cust_id INT NOT NULL,
	amount DECIMAL(6,2),
	p_loc INT NOT NULL,
```

```
      p_date DATE,
      r_loc INT NOT NULL,
      r_date DATE,
    PRIMARY KEY (reservation_id),
    FOREIGN KEY (VIN) REFERENCES cars(VIN),
    FOREIGN KEY (cust_id) REFERENCES customers(id),
    FOREIGN KEY (p_loc) REFERENCES locations(loc_id),
    FOREIGN KEY (r_loc) REFERENCES locations(loc_id)
);
```

ACTION OUTPUT



Action Output

	Time	Action	Response	Duration / Fetch Time
1	14:31:22	CREATE DATABASE crc	1 row(s) affected	0.112 sec
2	14:31:23	USE crc	0 row(s) affected	0.0018 sec
3	14:31:23	CREATE TABLE categories (cat_id INT NOT NULL AUTO_INCREMENT, cat_label VARCHAR(45), cat_desc VARCHAR(255...	0 row(s) affected	0.065 sec
4	14:31:23	CREATE TABLE cars (VIN VARCHAR(7) NOT NULL, car_desc VARCHAR(255), color VARCHAR(45), brand VARCHAR(...	0 row(s) affected	0.025 sec
5	14:31:23	CREATE TABLE locations (loc_id INT NOT NULL AUTO_INCREMENT, street VARCHAR(45), streetno VARCHAR(10), --...	0 row(s) affected	0.013 sec
6	14:31:23	CREATE TABLE phones (phone_id INT NOT NULL AUTO_INCREMENT, phone_no VARCHAR(15), loc_id INT NOT NULL,...	0 row(s) affected	0.021 sec
7	14:31:23	CREATE TABLE customers (id INT NOT NULL AUTO_INCREMENT, first_name VARCHAR(45), last_name VARCHAR(45),...	0 row(s) affected	0.018 sec
8	14:31:23	CREATE TABLE reservations (reservation_id INT NOT NULL AUTO_INCREMENT, VIN VARCHAR(7) NOT NULL, cust_id...	0 row(s) affected	0.032 sec

----------END OF SECTION 3----------

Section 4: Insert Statements

CODE

```
-- INSERTING DATA
USE crc;

INSERT INTO categories (cat_label, cat_desc) VALUES

("Compact", "Sedan-type car with 5 doors"),
("Convertible", "The roof of the car is retractable, hard-top or soft-top"),
("Jeep", "4X4, tall vehicle, usually suitable for off-road conditions"),
("Luxury", "Long town car, usually used for professional chauffer services"),
("SUV", "A mix between a sedan and a Jeep, medium-to-tall height, suitable for both
city and off-road conditions"),
("Hatchback", "Small car, with a flat back-side, usually suitable for low
consumption and convenient parking"),
("Pickup", "Pickup truck, with a large open or closed trunk, suitable for personal
or professional use");

-----------------------------------------------------------------------------------

INSERT INTO cars (VIN, car_desc, color, brand, model, cat_id, purch_date) VALUES

("ZTY4567", "Convertible with hardtop, leather seats and CD player", "Silver",
"Mercedes-Benz", "SLK200", 2, "2007-02-08"),
("ATB2646", "Professional and good-looking, full-extra", "Blue", "BMW", "160i", 1,
"2012-08-01"),
("IKP3998", "Stylish and eye-catching, with GPS", "Red", "Suzuki", "Swift", 6,
"2013-12-09"),
("IKA8788", "Low consumption and convenience, hybrid, with GPS", "Silver",
"Toyota", "Auris", 6, "2016-11-01"),
("IBN1220", "Luxurious and business-ready, leather seats, cruise control", "Black",
"BMW", "520i", 1, "2005-03-10"),
("IPK1002", "Easy handling, automatic, parking assistant", "Blue", "Opel", "Corsa",
6, "2011-11-17"),
("KMX3344", "Eye-catching and elegant, 360 parking assistant, bluetooth", "White",
"Nissan", "Juke", 5, "2016-02-01"),
("POO9821", "Convenience and off-road capabilities, leather seats, CD player",
"Orange", "Nissan", "Navara", 7, "2014-10-23"),
("IBN5786", "Luxurious and business-ready", "Black", "Mercedes-Benz", "S500", 4,
"2010-08-06"),
("YKP3668", "Hybrid, convenience, parking assistant, bluetooth", "Red", "Toyota",
"Auris", 6, "2017-06-06"),
("ZMP1210", "Low consumption with 5 doors, CD player", "Red", "Toyota", "Yaris", 6,
"2013-01-10"),
("IKP2221", "Parking assistant, 5 doors, automatic", "Red", "Opel", "Corsa", 6,
"2011-08-12"),
("IBT4312", "4X4 with up to 9 seats and off-road capatilities", "Red", "Jeep",
"Grand Cherokee", 3, "2009-01-10"),
("PIK5665", "Limousine with up to 9 luxurious leather seats and bar", "Red",
"Lincoln", "Town Car", 4, "2004-04-17");

-----------------------------------------------------------------------------------

INSERT INTO locations VALUES

("1","Hilpert Rapid","23","North Anastasia","NewYork","USA"),
("2","Donald Ways","286","Lake Toney","NewYork","USA"),
("3","Leann Trafficway","169","South Cara","NewJersey","USA"),
("4","Wiegand Views","99","Kuhlmanland","California","USA"),
("5","Ledner Turnpike","256","North Joshuah","California","USA"),
("6","Urban Mall","43","North Evalyn","Arizona","GB"),
("7","Wunsch Road","40","West Lorenz","Minnesota","IN"),
("8","Ludie Mountains","108","Port Kaitlyn","Tennessee","IE"),
```

```
("9","Aubrey Cliffs","186","Port Lessieborough","Missuri","CN"),
("10","Raleigh Cove","186","Lempitown","Nebraska","FR");
```

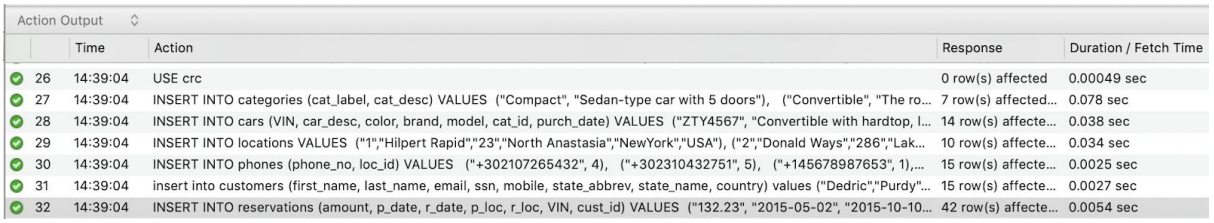
```
INSERT INTO phones (phone_no, loc_id) VALUES

("+302107265432", 4),
("+302310432751", 5),
("+145678987653", 1),
("+145678987654", 1),
("+442112368123", 2),
("+442112368124", 2),
("+507621848001", 3),
("+302107265433", 4),
("+302310432750", 5),
("+391329845421", 6),
("+421237898124", 7),
("+317901663925", 8),
("+206981797677", 9),
("+106541672231", 10),
("+106541672232", 10);
```

```
INSERT INTO customers (first_name, last_name, email, ssn, mobile, state_abbrev,
state_name, country) VALUES

("Dedric","Purdy","monty33@tillmanernser.net","737-73-2213","022-280-370972","MI","
Michigan","Cyprus"),
("Gayle","Ferry","rhoda76@bailey.com","931-13-2414","1-574-831-0280","NC","NorthCar
olina","Christmas Island"),
("Romaine","Gutkowski","xdubuque@schumm.com","323-33-2233","+85(1)810596397","MIS",
"Mississippi","Cook Islands"),
("Selena","Dach","orville84@gmail.com","862-26-2682","(787)403-08989","MA","Massach
usetts","Palau"),
("Dayana","Keefe","nadia87@murphy.com","646-64-4664","04551803452","HAW","Hawaii","
Montserrat"),
("Ryley","Weimann","spencer.rozella@hotmail.com","261-16-6212","308-383-911579","FL
O","Florida","Cook Islands"),
("Lois","Frami","rhoda21@hessel.info","713-15-2204","324-017-982855","MAR","Marylan
d","Singapore"),
("Kali","Monahan","edwina.auer@gmail.com","737-73-1234","760=111-3122","TEN","Tenne
ssee","Peru"),
("Ransom","Brown","nzemlak@hotmail.com","843-34-3821","858.373.739295","WYO","Wyomi
ng","Cambodia"),
("Jensen","Haag","schaefer.earnestine@gmail.com","907-79-0891","+07(4)621652845","T
EX","Texas","USA"),
("Jeramy","Reilly","jacobi.jodie@yahoo.com","725-52-2516","399.543.013362","MIS","M
ississippi","USA"),
("Edgardo","Wolf","jay02@yahoo.com","784-48-8471","(827)905-7283","NEV","Nevada","L
ebanon"),
("Jalen","Spencer","celine.blick@leschfritsch.com","910-21-1099","(365)321-0966","M
IN","Minnesota","USA"),
("Dimitri","Kon","beier.pearline@heaney.com","977-79-9905","653-476-5758","OKL","Ok
lahoma","USA"),
("Lester","Volkman","esmeralda12@koelpin.com","717-66-567","933-195-4371","GEO","Ge
orgia","USA");
```

```
INSERT INTO reservations (amount, p_date, r_date, p_loc, r_loc, VIN, cust_id)
VALUES
("132.23", "2015-05-02", "2015-10-10", 1, 1, "ZTY4567", 2),
("111.11", "2015-05-03", "2015-05-04", 2, 2, "IKA8788", 8),
("222.22", "2015-05-04", "2015-05-05", 3, 3, "ATB2646", 8),
```

```
("333.33", "2015-05-06", "2015-05-06", 1, 2, "IPK1002", 9),
("444.44", "2015-05-07", "2015-05-08", 3, 1, "ZTY4567", 11),
("300.20", "2015-03-14", "2015-03-14", 10, 9, "IKP3998", 7),
("29.10", "2015-05-20", "2015-05-22", 2, 3, "KMX3344", 15),
("1000.00", "2015-05-20", "2015-07-29", 4, 4, "YKP3668", 14),
("69.90", "2015-03-08", "2015-03-22", 8, 5, "KMX3344", 10),
("239.25", "2017-12-31", "2018-01-10", 6, 6, "IPK1002", 2),
("41.22", "2014-01-22", "2014-01-23", 5, 2, "ZTY4567", 1),
("22.33", "2015-05-09", "2015-05-31", 3, 7, "IKA8788", 3),
("38.26", "2015-10-13", "2015-10-15", 7, 7, "KMX3344", 6),
("77.88", "2015-10-22", "2015-10-27", 9, 1, "IBN1220", 5),
("99.66", "2015-05-23", "2015-05-29", 2, 5, "IKP3998", 13),
("34.12", "2015-10-24", "2015-10-28", 8, 1, "IPK1002", 4),
("667.99", "2015-03-19", "2015-03-20", 5, 7, "ZMP1210", 7),
("156.01", "2014-08-06", "2014-08-12", 1, 2, "POO9821", 12),
("39.29", "2015-01-07", "2015-01-11", 2, 8, "IKA8788", 11),
("45.55", "2015-01-08", "2015-01-10", 2, 2, "IKA8788", 9),
("21.18", "2015-03-02", "2015-03-04", 3, 4, "IBN5786", 2),
("109.57", "2015-08-01", "2015-08-15", 3, 3, "IBN5786", 15),
("124.56", "2014-05-15", "2014-05-23", 4, 4, "IBT4312", 6),
("543.22", "2014-05-24", "2014-05-08", 4, 5, "ZTY4567", 7),
("578.34", "2014-10-01", "2014-10-25", 4,8, "ZTY4567", 8),
("199.99", "2014-04-06", "2018-04-15", 5, 5, "ZMP1210", 2),
("201.02", "2014-04-14", "2014-04-20", 5, 5, "IBN1220", 1),
("202.03", "2014-10-08", "2014-10-19", 5, 4, "IBN1220", 13),
("68.71", "2018-07-11", "2018-07-14", 6, 6, "ATB2646", 11),
("78.81", "2015-08-25", "2015-08-30", 6, 7, "ATB2646", 5),
("88.91", "2014-10-02", "2014-10-07", 7, 7, "KMX3344", 8),
("15.05", "2015-07-08", "2010-07-08", 8, 8, "IPK1002", 11),
("14.13", "2015-06-10", "2015-06-11", 9, 9, "IKP3998", 2),
("19.20", "2015-06-30", "2015-07-01", 9, 9, "IKP2221", 3),
("248.84", "2015-04-16", "2015-04-29", 9,10, "IKP2221", 4),
("76.67", "2015-04-17", "2015-04-23", 9, 7, "IKP3998", 5),
("129.99", "2015-04-01", "2015-04-10", 9, 9, "IKP3998", 6),
("312.21", "2014-05-26", "2014-05-06", 10, 10, "KMX3344", 15),
("54.45", "2015-05-08", "2014-05-10", 10,10, "PIK5665", 14),
("34.45", "2014-08-09", "2014-08-11", 9,9, "PIK5665", 14),
("23.19", "2014-08-10", "2014-05-12", 7,7, "PIK5665", 13),
("41.68", "2014-08-14", "2014-08-17", 6,6, "PIK5665", 14);
```

ACTION OUTPUT



Action Output

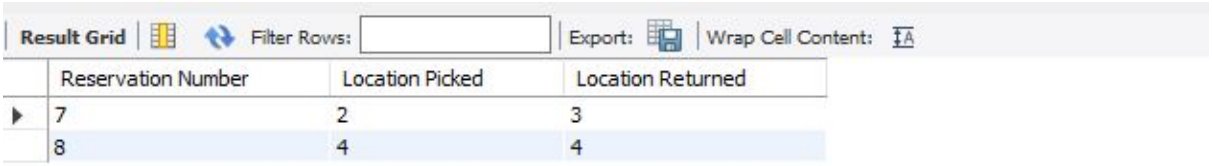
	Time	Action	Response	Duration / Fetch Time
26	14:39:04	USE crc	0 row(s) affected	0.00049 sec
27	14:39:04	INSERT INTO categories (cat_label, cat_desc) VALUES ("Compact", "Sedan-type car with 5 doors"), ("Convertible", "The ro...	7 row(s) affected...	0.078 sec
28	14:39:04	INSERT INTO cars (VIN, car_desc, color, brand, model, cat_id, purch_date) VALUES ("ZTY4567", "Convertible with hardtop, l...	14 row(s) affecte...	0.038 sec
29	14:39:04	INSERT INTO locations VALUES ("1","Hilpert Rapid","23","North Anastasia","NewYork","USA"), ("2","Donald Ways","286","Lak...	10 row(s) affecte...	0.034 sec
30	14:39:04	INSERT INTO phones (phone_no, loc_id) VALUES ("+302107265432", 4), ("+302310432751", 5), ("+145678987653", 1),...	15 row(s) affecte...	0.0025 sec
31	14:39:04	insert into customers (first_name, last_name, email, ssn, mobile, state_abbrev, state_name, country) values ("Dedric","Purdy"...	15 row(s) affecte...	0.0027 sec
32	14:39:04	INSERT INTO reservations (amount, p_date, r_date, p_loc, r_loc, VIN, cust_id) VALUES ("132.23", "2015-05-02", "2015-10-10...	42 row(s) affecte...	0.0054 sec

----------END OF SECTION 4----------

Section 5: SQL Queries

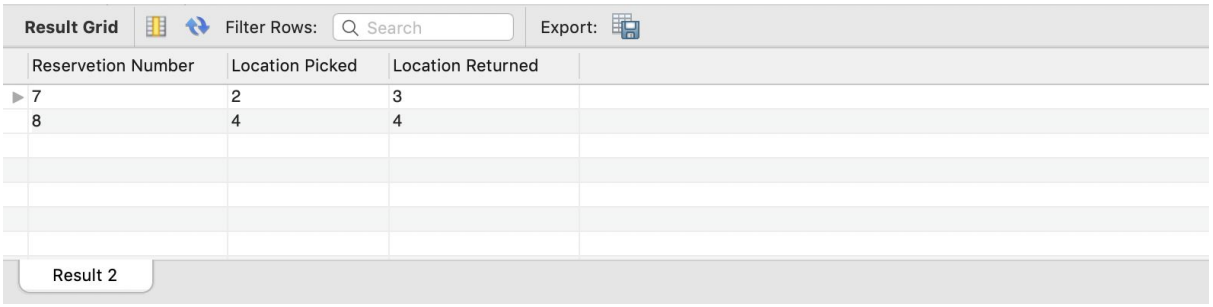
Question a: Show the reservation number and the location ID of all rentals on 5/20/2015

```
-- 1st solution (Assuming we want rentals that have 05/20/2015 as the
pickup date)

select reservation_id as 'Reservation Number',  p_loc as 'Location Picked',
r_loc as 'Location Returned'
from  reservations, locations
where p_date='2015-05-20' and p_loc=loc_id;
```



Reservation Number	Location Picked	Location Returned
7	2	3
8	4	4

```
-- 2nd option (If we want to include also Rentals that were completed on
05/20/2015)

select reservation_id as 'Reservation Number',  p_loc as 'Location Picked',
r_loc as 'Location Returned'
from  reservations, locations
where (p_date='2015-05-20' or r_date='2015-05-20') and p_loc=loc_id;
```

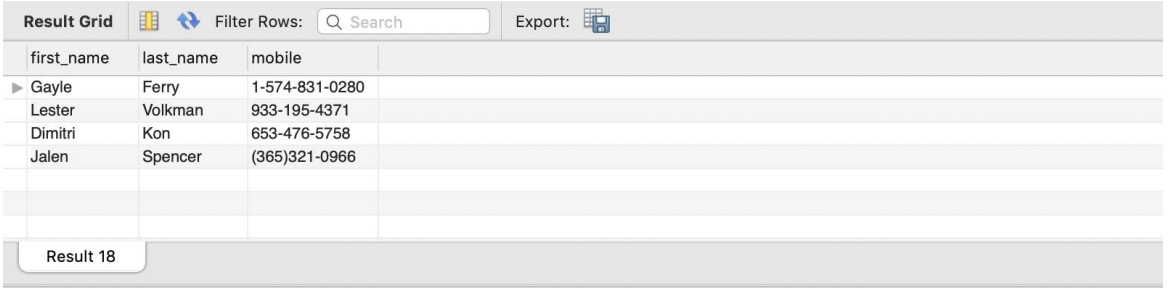


Reservetion Number	Location Picked	Location Returned
7	2	3
8	4	4

Result 2

Question b: Show the first and the last name and the mobile phone number of these customers that have rented a car in the category that has label = 'luxury'

```
-- If we want to see which customers have made more than one "Luxury"
reservation, we can remove the "distinct" operator

select distinct first_name, last_name, mobile
from reservations, cars, customers, categories
where categories.cat_label='Luxury' and reservations.cust_id=customers.id
and cars.VIN=reservations.VIN and cars.cat_id=categories.cat_id;
```

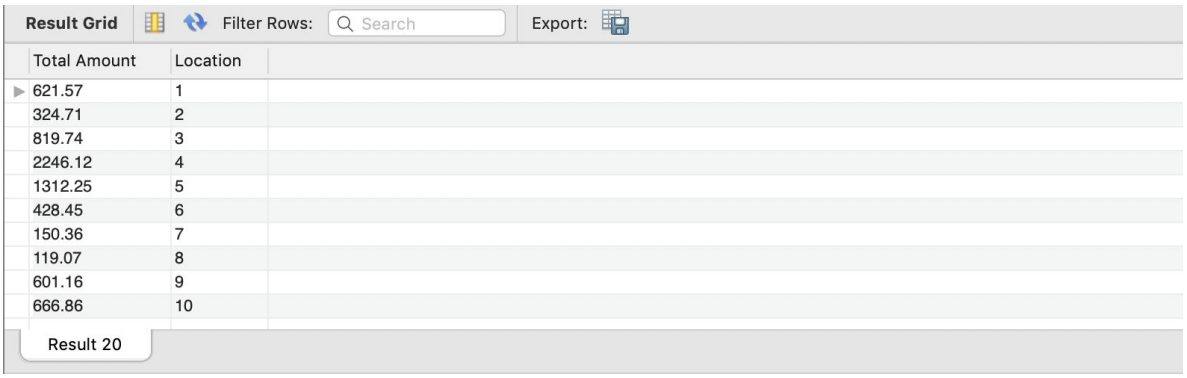


first_name	last_name	mobile
Gayle	Ferry	1-574-831-0280
Lester	Volkman	933-195-4371
Dimitri	Kon	653-476-5758
Jalen	Spencer	(365)321-0966

Result 18

Question c: Show the total amount of rentals per location ID (pick up)

```
select sum(amount) as 'Total Amount', p_loc as 'Location'
from reservations
group by p_loc;
```



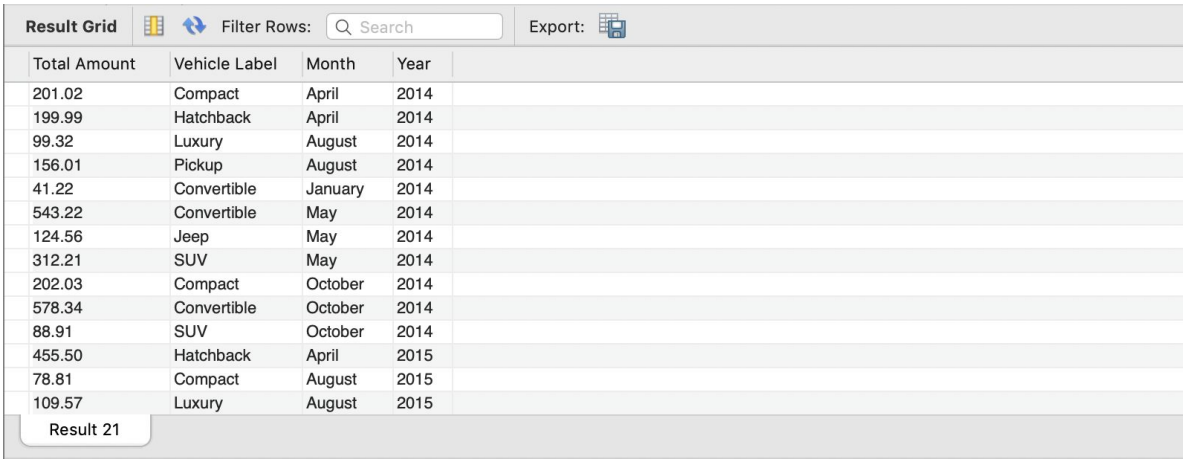
Result Grid | Filter Rows: Search | Export:

Total Amount	Location
621.57	1
324.71	2
819.74	3
2246.12	4
1312.25	5
428.45	6
150.36	7
119.07	8
601.16	9
666.86	10

Result 20

Question d: Show the total amount of rentals per car's category ID and month

```
-- We could also use cat_id instead of cat_label, but it would be less interpretable

select sum(amount) as 'Total Amount', cat_label as 'Vehicle Label',
monthname(p_date) as 'Month', extract(year from reservations.p_date) as
'Year'
from reservations, cars, categories
where reservations.vin=cars.vin and categories.cat_id=cars.cat_id
group by cat_label, Month, Year
order by year asc, monthname(p_date) asc;
```



Result Grid | Filter Rows: Search | Export:

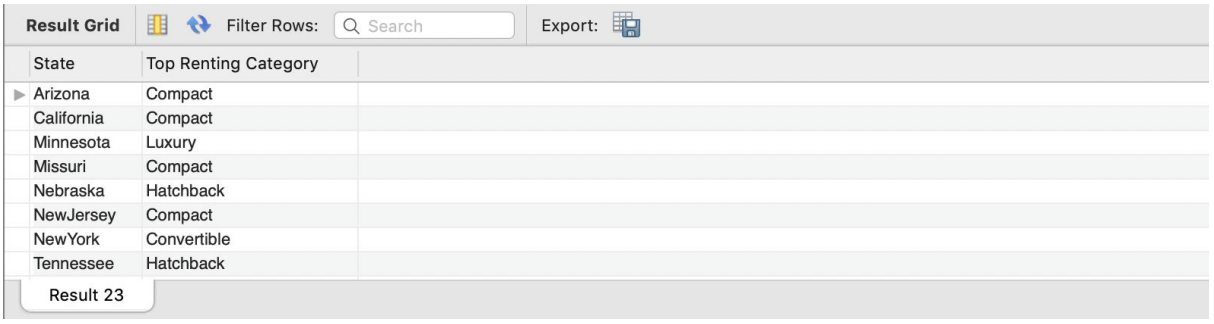
Total Amount	Vehicle Label	Month	Year
201.02	Compact	April	2014
199.99	Hatchback	April	2014
99.32	Luxury	August	2014
156.01	Pickup	August	2014
41.22	Convertible	January	2014
543.22	Convertible	May	2014
124.56	Jeep	May	2014
312.21	SUV	May	2014
202.03	Compact	October	2014
578.34	Convertible	October	2014
88.91	SUV	October	2014
455.50	Hatchback	April	2015
78.81	Compact	August	2015
109.57	Luxury	August	2015

Result 21

Question e: For each rental"s state (pick up) show the top renting category

```
select State, Label as 'Top Renting Category'
from (
select lstate as State, count(cars.cat_id) as TotalCount,
categories.cat_label as Label
from reservations, locations, cars, categories
```

```
where reservations.p_loc=locations.loc_id and cars.VIN=reservations.VIN and
categories.cat_id=cars.cat_id
group by State, Label
order by State
) as question_5
group by State;
```



Result Grid | Filter Rows: Search | Export:

State	Top Renting Category
Arizona	Compact
California	Compact
Minnesota	Luxury
Missuri	Compact
Nebraska	Hatchback
NewJersey	Compact
NewYork	Convertible
Tennessee	Hatchback

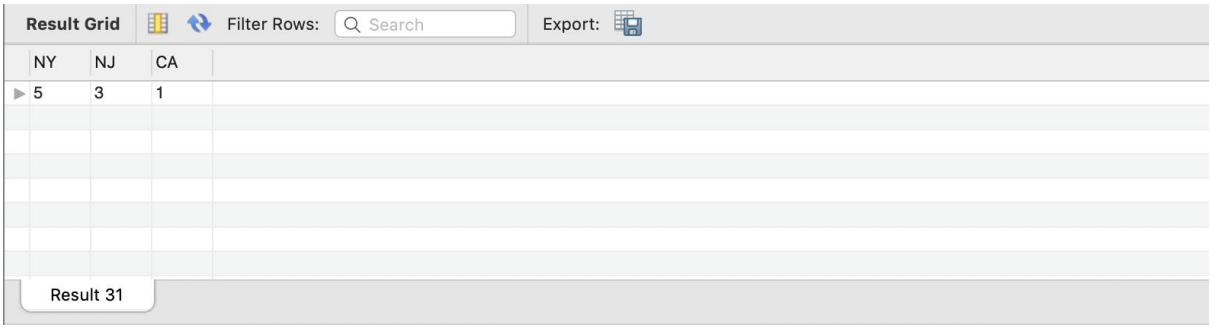
Result 23

-- ALTERNATIVE USING create view

```
create view question_5 (State, TotalCount, Label) as
select lstate as State, count(cars.cat_id) as TotalCount,
categories.cat_label as Label
from reservations, locations, cars, categories
where reservations.p_loc=locations.loc_id and cars.VIN=reservations.VIN and
categories.cat_id=cars.cat_id
group by State, Label
order by State;

select State, Label as 'Top Renting Category'
from question_5
group by State;
```

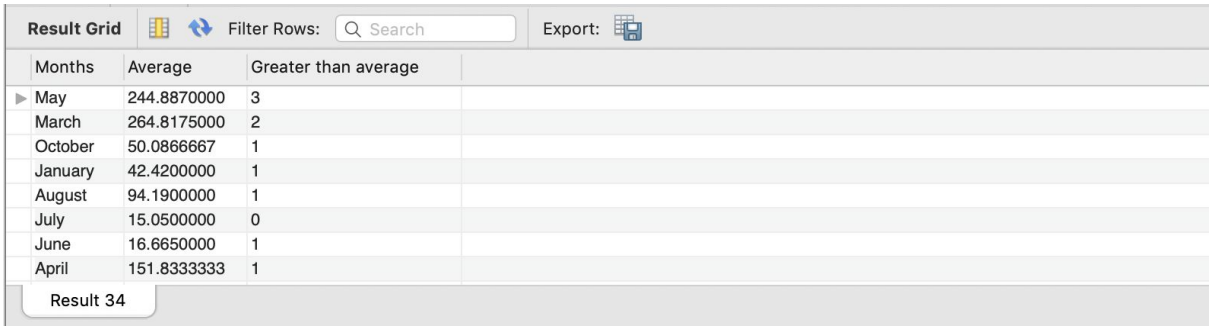
Question f: Show how many rentals there were in May 2015 in „NY", „NJ" and „CA" (in three columns)

```
create view rent_pickup1(Receipt, State, Country) as
select count(reservation_id) as Receipt, lstate, lcountry
from reservations, locations
where (p_date like '2015-05%' or r_date like '2015-05%') and
(lstate='NewYork' or lstate='NewJersey' or lstate='California')
and p_loc = loc_id
group by lstate, lcountry;

select *
from (
select sum(NY) as NY, sum(NJ) as NJ, sum(CA) as CA
      from (
      select max(case when State='NewYork' then receipt end) as NY, max(case
when State='NewJersey' then receipt end) as NJ, max(case when
State='California' then receipt end) as CA
      from rent_pickup1
      ) as rent_pickup2
) as rent_pickup3;
```



Question g: For each month of 2015, count how many rentals had amount greater than this month's average rental amount

```
create view question_7(Reservation, Amount, Months) as
select reservation_id, amount, monthname(p_date) as Month
from reservations
where p_date like '2015%';

select t1.Months, t2.Average,
count(case when Amount > Average then 1 end) as 'Greater than average'
from question_7 as t1
join (
select Months, avg(Amount * 1.0) as Average
from question_7
group by Months
) as t2 on t1.Months = t2.Months
group by Months;
```



Question h: For each month of 2015, show the percentage change of the total amount of rentals over the total amount of rentals of the same month of 2014

```
select month2015 as 'Month',
concat(round((reservation2015-reservation2014)/reservation2014*100),'%') as
Percentage_Change
from (
      select count(reservation_id) as reservation2015, monthname(p_date) as
Month2015
      from reservations
      where year(p_date)='2015'
      group by Month2015
) as Year_2015 join (
      select count(reservation_id) as reservation2014, monthname(p_date) as
Month2014
      from reservations
```

```
      where year(p_date)='2014'
      group by Month2014
) as Year_2014
where Year_2015.Month2015=Year_2014.Month2014;
```

Result Grid | Filter Rows: Search | Export:

Month	Percentage_Change
May	233%
October	0%
January	100%
August	-50%
April	50%

Result 35

Question i: For each month of 2015, show in three columns: the total rentals' amount of the previous months, the total rentals" amount of this month and the total rentals' amount of the following months

```
create view question_9 as
select sum(amount) as Total_Amount, month(p_date) as Month_of_2015
from reservations
where year(p_date)='2015'
group by Month_of_2015
order by Month_of_2015;

select amounts_table.Month_2015, amounts_table.Previous_Months as 'Previous
Months Total Amount', amounts_table.Current_Month_Amount,
sum(NextMonth.Total_Amount) as 'Next Months Total Amount'
from (select question_9.Month_of_2015 AS  Month_2015,
sum(Previous.Total_Amount) as Previous_Months, question_9.Total_Amount as
Current_Month_Amount
from question_9
left join question_9 as Previous on question_9.Month_of_2015 >
Previous.Month_of_2015
group by Month_2015 ) as amounts_table
left join question_9 as NextMonth on amounts_table.Month_2015 <
NextMonth.Month_of_2015
group by Month_2015;
```

Result Grid | Filter Rows: Search | Export:

Month_2015	Previous Months Total Amount	Current_Month_Amount	Next Months Total Amount
1	NULL	84.84	4350.66
3	84.84	1059.27	3291.39
4	1144.11	455.50	2835.89
5	1599.61	2448.87	387.02
6	4048.48	33.33	353.69
7	4081.81	15.05	338.64
8	4096.86	188.38	150.26
10	4285.24	150.26	NULL

Result 36

----------END OF SECTION 5----------

Section 6: Questions 4 & 5 using R (connected to MySQL)

Notes: How we solved various problems connecting R to MySQL

Trying to connect MySQL with R and then populate the table "customers" with data from the .csv file, we encountered 2 problems.

Problem #1:
Problem loading the "caching_sha2_password" plugin, due to the default settings of the newest MySQL version. This problem can be solved with 2 methods:

A) Perform an initiation of the MySQL server and configure it to use the "Legacy Password Encryption" option.
or
B) Open console > connect to MySQL > Run:

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY
'newrootpassword';
```

Problem #2:
Problem with the command `dbWriteTable()`, because, MySQL 8.0 does not allow user access to data loads from local sources, in contrast with MySQL 5.6.

The error we got was:

```
Error in .local(conn, statement, ...) :
  could not run statement: The used command is not allowed with this MySQL version
```

This was solved, by opening the console, connecting to mysql and running:

```
mysql> SET GLOBAL local_infile = true;
mysql> SHOW GLOBAL VARIABLES LIKE 'local_infile';
```

Question 4: You are given a csv file called “temp.csv” (comma delimited). Using the programming language of your choice, open the file, connect to the database, and populate the table storing customers in your schema (insert). The file is in the format SSN, First Name, Last Name, mobile phone number, email, ID, state, country.

```
# install.packages("DBI")
# install.packages("RMySQL")

library(DBI)
library(RMySQL)

customersdf <- read.csv(file="Assignment_1_Customers.csv", header=TRUE, sep=",")

#You need to input your own database name, username and password here
mydb <- dbConnect(dbDriver("MySQL"), user = "root", password="f2821803", dbname =
"crc", host="localhost", port=3306)

dbWriteTable(mydb, value = customersdf, row.names = FALSE, name = "customers",
append = TRUE)
dbReadTable(mydb, "customers")


### DISCONNECT FROM DATABASE ###
dbDisconnect(mydb)
```

RESULT

id	first_name	last_name	email	ssn	mobile	state_abbrev	state_name	country
16	Glyn	Targett	gtargettf@ycombinator.com	112-12-0388	411-395-9921			Philippines
17	Cornall	Kersey	ckerseyg@mashable.com	667-54-8721	520-743-6065	PHG	Pahang	Malaysia
18	Ly	MacRorie	lmacrorieh@businesswire.com	700-29-5468	462-199-1937			New Zealand
19	Luciano	Saph	lsaphi@blinklist.com	837-26-3248	282-291-7295			China
20	Jerrine	Cornes	jcornesj@arizona.edu	620-68-7180	681-330-8508			Panama
21	Elvira	Kares	ekaresk@yellowbook.com	564-01-9327	669-724-3148			Indonesia
22	Caryl	Louisot	clouisotl@timesonline.co.uk	591-62-5271	148-167-9321			Nigeria
23	Dawna	Passler	dpasslerm@springer.com	418-12-8580	483-918-6818			China
24	Ilario	Nodin	inodinn@harvard.edu	438-61-6597	970-560-0015			Russia
25	Marilin	Daffern	mdafferno@miibeian.gov.cn	140-86-1744	562-754-4635			Thailand
26	Hilda	Stoven	hstovenp@php.net	795-78-0289	781-632-9908	AB	Alberta	Canada
27	Candide	Perkinson	cperkinsonq@wordpress.org	430-31-0508	360-254-8926			Madagascar
28	Lenna	Rickesies	lrickesiesr@economist.com	889-15-8082	204-401-9399			Czech Republic
29	Sascha	Quene	squenes@sogou.com	588-57-4018	971-774-2960			Indonesia
30	Rochette	Janzen	rjanzent@google.com.br	108-71-7814	875-878-9337			Thailand
31	Susanna	Mahady	smahadyu@biglobe.ne.jp	741-61-9051	161-270-4075			Brazil

customers 2

3 16:17:38 select * from customers LIMIT 0, 1000 1000 row(s) returned

Question 5: Using the programming language of your choice, connect to the database and implement query (i) above – without using GROUP BY SQL statements, i.e. you are only allowed to use SELECT...FROM...WHERE. Best implementation gets a bonus :)

```
 install.packages("DBI")
# install.packages("RMySQL")

library(DBI)
library(RMySQL)

#You need to input your own database name, username and password here
mydb <- dbConnect(dbDriver("MySQL"), user = "root", password="f2821803", dbname =
"crc", host="localhost", port=3306)


monthlist = list()
for (i in 1:12) {
       query<-paste("select sum(amount) from reservations where year(p_date)=2015
AND month(p_date)=", i,";", sep="")
       sendquery <- dbSendQuery(mydb, query)
       data <- fetch(sendquery, n=1)
       dbClearResult(sendquery)
       data$month<- i
       monthlist[[i]] <- data
}
MonthTotal= do.call(rbind, monthlist); names(MonthTotal)<-c("MonthlyTotal",
"Month")
MonthTotal[is.na(MonthTotal)] <- 0

PreviousMonths<-0
NextMonths<-sum(MonthTotal$MonthlyTotal)
Totals = list()
Totals <- NULL
for (i in 1:12) {
       CurrentMonth<-MonthTotal[i,1]
       NextMonths<-NextMonths-CurrentMonth
       Totals[[i]]<-c(i, PreviousMonths, CurrentMonth, NextMonths)
       PreviousMonths <-PreviousMonths + MonthTotal[[i,1]]
}
Totals <- data.frame(do.call(rbind, Totals)); names(Totals)<-c("Month", "Previous
Months", "Current Month", "Next Months");
print(Totals)

### DISCONNECT FROM DATABASE ###
dbDisconnect(mydb)
```

RESULT:

```
> print(Totals)
   Month Previous Months Current Month Next Months
1      1            0.00         84.84     4350.66
2      2           84.84          0.00     4350.66
3      3           84.84       1059.27     3291.39
4      4         1144.11        455.50     2835.89
5      5         1599.61       2448.87      387.02
6      6         4048.48         33.33      353.69
7      7         4081.81         15.05      338.64
8      8         4096.86        188.38      150.26
9      9         4285.24          0.00      150.26
10    10         4285.24        150.26        0.00
11    11         4435.50          0.00        0.00
12    12         4435.50          0.00        0.00
>
```